UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/596,257	06/16/2000	Wendi L. Nusbickel	6169-155	3198

7590 09/22/2003

Gregory A Nelson
Quarles & Brady LLP
222 Lakeview Avenue Fourth Floor
P O Box 3188
Weast Palm Beach, FL 33402-3188

EXAMINER
CAO, DIEM K

ART UNIT	PAPER NUMBER
2126	7

DATE MAILED: 09/22/2003

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/596,257	NUSBICKEL, WENDI L.	
	Examiner	Art Unit	
	Diem K Cao	2126	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
- Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *03 July 2003* .

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-16* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-16* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

11)☐ The proposed drawing correction filed on _____ is: a)☐ approved b)☐ disapproved by the Examiner.

If approved, corrected drawings are required in reply to this Office action.

12)☐ The oath or declaration is objected to by the Examiner.

Priority under 35 U.S.C. §§ 119 and 120

13)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____ .

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

14)☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. § 119(e) (to a provisional application).

a) ☐ The translation of the foreign language provisional application has been received.

15)☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. §§ 120 and/or 121.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449) Paper No(s) _____ .
4) ☐ Interview Summary (PTO-413) Paper No(s). _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: .

DETAILED ACTION

1. This Office Action is in response to the Amendment filed on 7/3/2003.

2. Claims 1-16 remain in the application. Applicant has amended claims 4 and 12.

Claim Rejections - 35 USC § 103

3. The text of those sections of Title 35, U.S. Code not included in this action can be found in a prior Office action.

4. Claims 1-2, 4-5, 9-10 and 12-13 are rejected under 35 U.S.C. 103(a) as being unpatentable over Riehle (The Event Notification Pattern - Integrating Implicit Invocation with Object-Orientation).

As to claim 1, Riehle teaches (pages 4-9) a Notifier object (Subject object, StateChange object), a Notifier class (Subject class, StateChange class), list of Listener objects (Observer objects), a Listener class (Observer class), events (event), the Listener object defines a method (operation) to be called when the occurrence of the event (provides the event stubs ... in case of invocation; page 6), the Listener objects enabled to be callable from the Notifier object (A state change object distributes ... an observer), a Listener object stub for the Listener object (EventStub), the Listener object stub configured to be added to the list of Listener objects in the Notifier object (StateChange offers ... via event stub object), the Listener object stub further configured to remotely call the defined method in the Listener object (An event stub forward ... an observer) in response to receiving notification of an event from the Notifier object (A StateChange object distributed ... to all its event stubs), upon the event occurrence, the Notifier object traverse the list of Listener objects and can notify the Listener object stub thereby creating a remote call to the defined method in the Listener object (If a subject changes its state ... to all

its event stubs).

However, Riehle does not explicitly teach the Notifier object in a client application for execution in a first process address space and the Listener object in a server application for execution in a second process address space. Riehle teaches the Event Notification Pattern can be used in the object-oriented distributed system (Both the Event Notification ... distributed system ... different respects; page 1), and Notifier object and Listener object are in different process address spaces (IACEventLink serves to make the notification transparent to process boundaries, the purpose of an event chain is to forward ... different process ... local observers; page 8).

It would have been obvious to one of ordinary skill in the art to modify the teaching of Riehle to have the Notifier object in a client application and Listener object in a server application because it provides a method to manage inter-object-dependencies in client/server application.

As to claim 2, Riehle does not explicitly teach the Notifier and the Listener classes are Java classes, and the first and second process address spaces are in the first and second Java Virtual Machine. Riehle teaches the Event Notification Pattern can be apply to object-oriented distributed system (Both the Event Notification ... distributed system ... different respects; page 1), and Notifier object and Listener object are in different process address spaces (IACEventLink serves to make the notification transparent to process boundaries, the purpose of an event chain is to forward ... different process ... local observers; page 8). It would have been obvious to use Java language to the system of Riehle because Java is neutral platform programming language, and inherently, the client and server applications are executed in different process spaces in the Java Virtual Machine.

As to claim 9, it corresponds to the method claim of claim 1, except it is a computer product for establish location transparent event handling.

As to claim 4, Riehle teaches (pages 4-9) an instance (Subject object, StateChange object) of a Notifier class (Subject class, StateChange class), a list of Listerner objects (Observer objects) to be notified upon an event occurrence (event), an instance (Observer object) of a Listener class (Observer class), the Listener instance having a method (method) to be called upon the occurrence of the event (provides the event stubs ... in case of invocation; page 6), the Listener instance enabled to be callable from the Notifier instance (A state change object distributes ... an observer), wherein the Notifier instance and the Listener instance are configured to perform location transparent event handling (This pattern lets developers decouple ... event link objects; page 3, section 2.2), inserting a Listener object stub in the list of Listener object in the Notifier instance (StateChange offers ... via event stub object), the Listener object stub configured to remotely call the defined method in the Listener instance (An event stub forward ... an observer), receiving an event occurrence in the Notifier instance (state change objects), responsive to receiving the event occurrence (a subject changes state), traversing the list of Listener objects and passing the event to the Listener object stub (a state change object distributes the notification to all its event stub), creating in the Listener object stub a remote call to the defined method in the Listener instance (an event stub forward a notification to its owner), and executing the defined method in the Listener instance (change in Observer object, inherently).

However, Riehle does not teach an instance of a Notifier class in a first address space, and an instance of a Listener object in a second address space. In different embodiment, Riehle

teaches Notifier object and Listener object are in different process address spaces (IACEventLink serves to make the notification transparent to process boundaries, the purpose of an event chain is to forward ... different process ... local observers; page 8). It would have been obvious to modify the system of Riehle to implement both embodiments because it provides a method to manage inter-object-dependencies in the distributed application.

As to claim 12, it is rejected under the same ground as of claim 4.

As to claims 5, 10, and 13, see rejection of claim 2 above.

5. Claims 3, 6-8, 11, and 14-16 are rejected under 35 U.S.C. 103(a) as being unpatentable over Riehle (The Event Notification Pattern - Integrating Implicit Invocation with Object-Orientation) in view of OMG (COM/CORBA Interworking RFP - Part A) further in view of Sun Microsystems (Remote Method Invocation Specification).

As to claim 6, Riehle does not explicitly teach the Listener object stub is generated in an RMI compilation process. Riehle teaches the Listener object is the owner of the Listener object stub (An event stub ... its owner, an observer; page 6), and the forwarding process might be implemented using a remote procedure call (A class IACEventLink ... boundaries; page 8). OMG teaches a CORBA object can subscribe to and handle events through its CORBA proxy, and the CORBA proxy relays the event back to the real CORBA object (When the Subscribing Object is a CORBA object section; page 49). Sun teaches the object stub is generated in an RMI compilation process (Stubs are generated using the rmic compiler; Type Equivalency of Remote Objects with Local Stub section). It would have been obvious the Java object could subscribe and handle events through its Java stub because Java can be implemented in distributed system using remote method invocation to invoke object on different process space.

As to claim 7, Riehle does not explicitly teach registering the Listener instance with an RMI registry, the RMI registry executing in a third Java Virtual Machine, the Notifier instance retrieving a reference to the registered Listener instance when inserting the Listener object stub to the list of Listener objects. Sun teaches (Registry Interfaces section) registering the Listener instance with an RMI registry (The RMI system ... by simple names), the RMI registry executing in a third Java Virtual Machine (Any server process can ... standalone), the Notifier instance retrieving a reference to the registered Listener instance when inserting the Listener object stub to the list of Listener objects (A simple bootstrap name server ... particular host and port; Locating Remote Objects section).

As to claim 8, Riehle as modified by Sun teach the step of creating in the Listener object stub remotely calls the defined method in the Listener instance through the retrieved reference upon receiving the event from the Notifier instance (Collaborations of Riehle; page 6 and Locating Remote Objects section; Sun Microsystems)

As to claims 3 and 11, see rejections of claims 6-8 above.

As to claims 14-16, see rejections of claims 6-8 above.

Response to Arguments

Claims 1 and 9

As to Applicant's arguments (page 9) regarding Riehle does not teach placing a notifier object and a listener object in separate address space. Applicant also argues that the interdependent software objects could be operated only in a tightly integrated environment characteristic of an environment having a single address space because of relationship between them (page 10). Applicant further argues that by placing the Notifier object in the client and

Listener object in a server violates the basic computing principle by requiring an arrangement where client behavior automatically results in server changes. However, Riehle clearly teaches the notifier object and listener object could be place in different address space (IACEventLink serves to make the notification transparent to process boundaries, the purpose of an event chain is to forward ... different process ... local observers; page 8). As to Applicant's argument regarding violating the basic computing principle, Examiner does not agree because one of ordinary skill in the art would know that in the distributed application, change in client application does make change in the server application, for example, user submits changes regarding user's information, which need to be updated in the server. Thus, Riehle teaches and suggests all the limitations of claims 1 and 9.

Claims 2, 5, 10 and 13

As to Applicant's arguments (pages 11-12) regarding Riehle does not teach the placement of a notifier and a listener within different JVMs, and Riehle maintains inter-object dependencies that would not exist across different JVMs. Applicant further argues that no implicit parameter passing/ memory sharing/ or process sharing occurs between two different JVMs. However, as discuss above regarding claims 1 and 9, Riehle teaches the notifier object and the listener object exist in different address space. Riehle also suggest the pattern can be used in distributed system (page 1). Furthermore, one of ordinary skill in the art knows the advantage of Java programming language wherein the language is neutral platform and Java Virtual Machine is needed in each system in order to execute the Java programs, and Java is used in distributed applications. Based on the teaching of Riehle and knowledge of one ordinary skill in the art, placing a notifier object and a listener object in different JVMs is taught.

Claims 4 and 10

As to Applicant's arguments (page 12) regarding Riehle does not disclose a method that calls remotely located procedures in response to a detected event. However, Riehle teaches IACEventLink servers to make the notification transparent to process boundaries, i.e., in different address space (page 8), and EventStub call an operation of the Observer object in case of event change (statechange). Thus, Riehle teaches the above limitation.

Applicant further argues that Riehle does not teach a method where the Notifier instance and the Listener instance are configured to perform location transparent event handling. This limitation is newly added and also taught by Riehle (see rejection of claim 4 above).

Claims 3, 6, 11, and 14

As to Applicant's arguments (page 12) regarding the teaching of OMG should not be combined with Riehle because Riehle teaches away from Riehle. However, OMG is used to teach the Corba object can subscribe and handle events published by OCXs through its proxy (page 49).

Applicant also argues (page 12-14) that the RMI model fails to include event handling procedure, and goes on to explain the different between RMI and CORBA technology. As discuss above, the Corba object can subscribe and handle events through its proxy. Also, one of ordinary skill in the art knows RMI generates stub that executes on the client side, and RMI stub equivalent to Corba proxy. In a distributed application written in Java, client invokes a method on the stub of a server object, which in turn, the stub invokes a remote method call on its server object. Thus, based on the teaching of OMG and knowledge of RMI, one of ordinary skill in the art would implement the Listener stub as an RMI stub.

Conclusion

6. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Diem K Cao whose telephone number is (703) 305-5220. The examiner can normally be reached on Monday - Thursday, 9:00AM - 5:00PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, John Follansbee can be reached on (703) 305-8498. The fax phone numbers for the organization where this application or proceeding is assigned are (703) 308-6296 for regular communications and (703) 305-9731 for After Final communications.

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the receptionist whose telephone number is (703) 305-3900.

Any response to this action should be mailed to:
Commissioner for Patents
PO Box 1450
Alexandria, VA 22313-1450

Or fax to:

- AFTER-FINAL faxes must be signed and sent to (703) 746-7238.
- OFFICIAL faxes must be signed and sent to (703) 746-7239.
- NON-OFFICIAL/DRAFT faxes should not be signed, please send to (703) 746-7140.

Diem Cao
September 15, 2003

JOHN FOLLANSBEE
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100